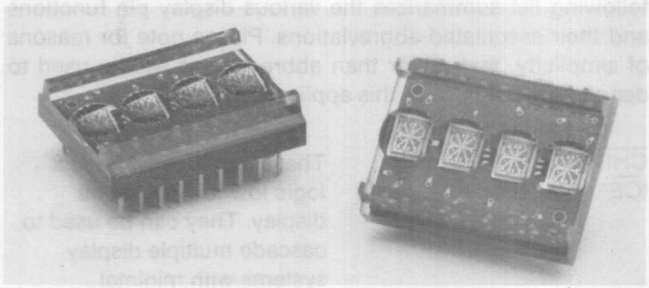
HEWLETT PACKARD

APPLICATION NOTE 1026

Designing with Hewlett-Packard's HPDL-2416 Smart Display

TABLE OF CONTENTS



INTRODUCTION

Hewlett-Packard's smart alphanumeric display, the HPDL-2416, is designed to minimize user design worries. Each HPDL-2416 displays four alphanumeric characters utilizing an on-board CMOS IC. The display is able to handle character storage, character decoding, multiplexing, segment driving and refreshing. Designers should treat the HDPL-2416 as a four-byte RAM, whose purpose is to store and display 64 character ASCII data using a sixteen-segment character font as shown in Figure 1.

This note is intended to serve as a design and application guide for users of the HPDL-2416. The information presented will cover: electrical description, electrical design considerations, interfacing to microprocessors, preprogrammed message systems, mechanical and electrical handling, and contrast enhancement.

BITS	D_3 D_2 D_1 D_0	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
D_6 D_5 D_4	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 1 0	2	(space)	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0 1 1	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
1 0 0	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1 0 1	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_

Figure 1. HPDL-2416 ASCII Character Set

ELECTRICAL DESCRIPTION

Figure 2 shows the internal block diagram of the HPDL-2416. It consists of two parts: the display LEDs and the CMOS IC. The CMOS IC is designed to operate from −20° C to +70° C; over this temperature range the inputs operate on standard TTL voltages (V_{IH} = 2.0 V min., V_{IL} = 0.8 V max.). The IC can store a maximum of four characters in ASCII memory and four cursors in CURSOR memory. Every location in ASCII memory stores one 7 bit ASCII character; each CURSOR memory location contains a single bit that denotes the presence or absence of a cursor. (All segments will be ON when a cursor is displayed.) Each location in memory may be randomly accessed. After a character has been written to memory the IC decodes the ASCII data, drives the display and refreshes it without any external hardware or software.

The HPDL-2416 uses 18 pins to control the CMOS IC. Figure 2 shows the effect these inputs have on the display. The following list summarizes the various display pin functions and their associated abbreviations. Please note for reasons of simplicity, text rather than abbreviations will be used to denote pin functions in this application note.

$\overline{\text{CHIP ENABLE}}$ INPUTS ($\overline{CE_1}$ and $\overline{CE_2}$, pins 1 and 2)	These inputs must be at a logic low to write to the display. They can be used to cascade multiple display systems with minimal additional hardware.
$\overline{\text{CLEAR}}$ ($\overline{\text{CLR}}$ pin 3)	ASCII data will be removed from the ASCII memory after the $\overline{\text{CELAR}}$ input has been held at a logic low for ms. CURSOR data is unaffected by the $\overline{\text{CLEAR}}$ input.
CURSOR ENABLE INPUT (CUE pin 4)	The IC uses CUE to determine whether it displays the ASCII memory or the CURSOR memory. (1 = CURSOR, 0 = ASCII).
$\overline{\text{CURSOR SELECT}}$ INPUT (CU pin 5)	This input is used by the IC to determine whether data is stored in ASCII memory or CURSOR memory. (1 = ASCII, 0 = CURSOR).
$\overline{\text{WRITE}}$ ($\overline{\text{WR}}$ pin 6)	Data is written into the display when the $\overline{\text{WR}}$ input is low and the display has been selected.
ADDRESS INPUTS (A_1 and A_0, pins 8 and 7)	Each location in memory has a distinct address. ADDRESS inputs enable the designer to select a specific location in memory to store data. Address 00 accesses the far right display location. Address 11 accesses the far left location.
V_{CC} and GND (pins 9 and 10)	These pins supply power to the display
DATA INPUTS (D_0-D_6, pins 11-17)	Seven bit ASCII data is entered into the ASCII memory via the DATA inputs. CURSOR data is stored in the CURSOR memory when the $\overline{\text{CU}}$ input is a logic low. D_0 determines if a CURSOR is stored or removed from CURSOR memory. (1 = store cursor, 0 = remove cursor)
$\overline{\text{BLANKING}}$ (BL pin 18)	The $\overline{\text{BLANKING}}$ input can be used to create a flashing display or to blank the display without clearing the ASCII memory. This input inhibits the IC segment drivers and the display $\overline{\text{CLEAR}}$ ($\overline{\text{CLR}}$) pin.

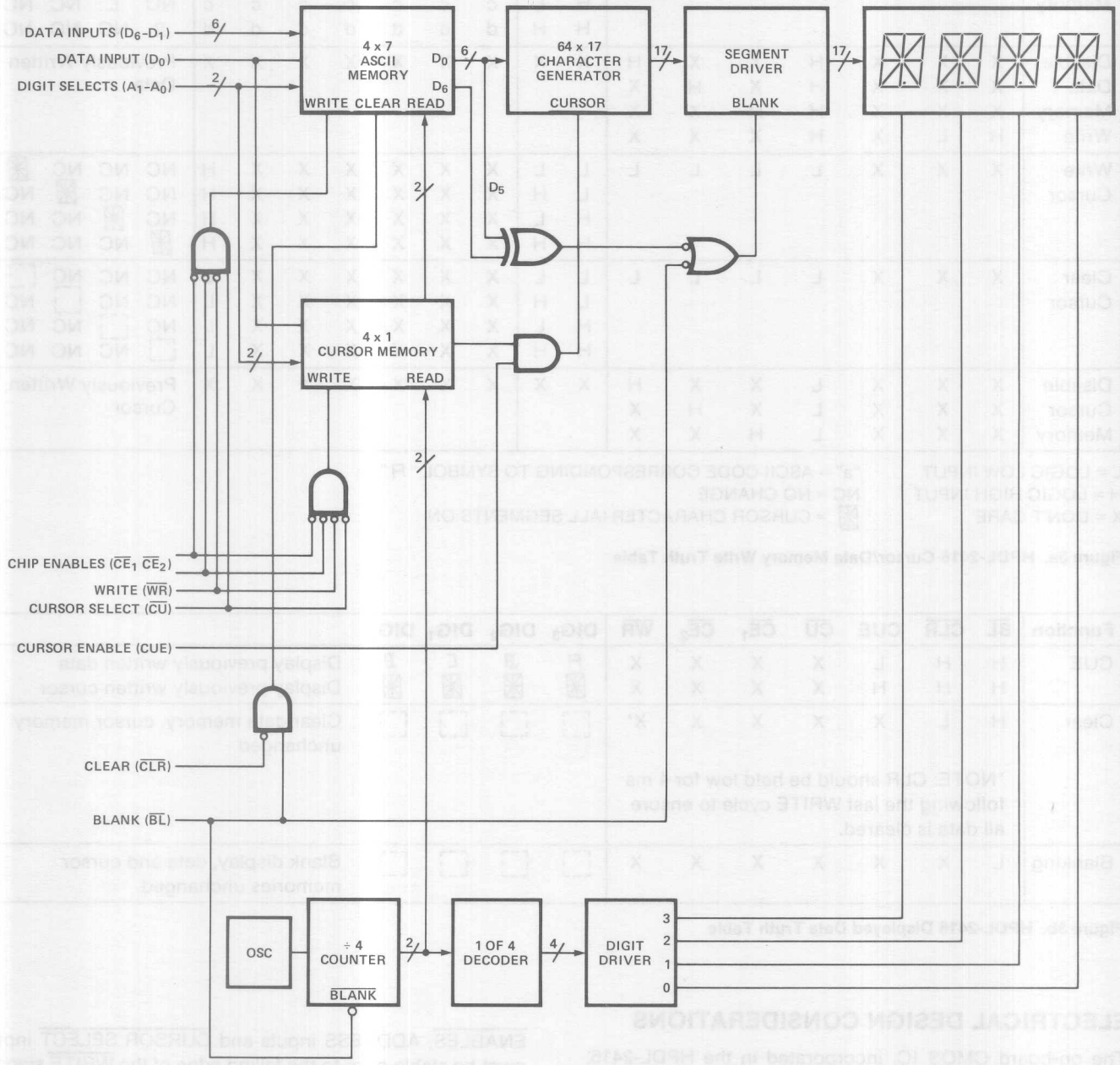


Figure 2. HPDL-2614 Internal Block Diagram

Function	$\overline{BL}$	$\overline{CLR}$	CUE	$\overline{CU}$	$\overline{CE_1}$	$\overline{CE_2}$	$\overline{WR}$	A_1	A_0	D_6	D_5	D_4	D_3	D_2	D_1	D_0	DIG_3	DIG_2	DIG_1	DIG_0
Write Data Memory	L	X	X	H	L	L	L	L	L	a	a	a	a	a	a	a	NC	NC	NC	A
	X	H	X	H	L	L	L	L	H	b	b	b	b	b	b	b	NC	NC	B	NC
								H	L	c	c	c	c	c	c	c	NC	C	NC	NC
								H	H	d	d	d	d	d	d	d	D	NC	NC	NC
Disable Data Memory Write	X	X	X	H	X	X	H	X	X	X	X	X	X	X	X	X	Previously Written Data			
	X	X	X	H	X	H	X													
	X	X	X	H	H	X	X													
	H	L	X	H	X	X	X													
Write Cursor	X	X	X	L	L	L	L	L	L	X	X	X	X	X	X	H	NC	NC	NC	▩
								L	H	X	X	X	X	X	X	H	NC	NC	▩	NC
								H	L	X	X	X	X	X	X	H	NC	▩	NC	NC
								H	H	X	X	X	X	X	X	H	▩	NC	NC	NC
Clear Cursor	X	X	X	L	L	L	L	L	L	X	X	X	X	X	X	L	NC	NC	NC	⬚
								L	H	X	X	X	X	X	X	L	NC	NC	⬚	NC
								H	L	X	X	X	X	X	X	L	NC	⬚	NC	NC
								H	H	X	X	X	X	X	X	L	⬚	NC	NC	NC
Disable Cursor Memory	X	X	X	L	X	X	H	X	X	X	X	X	X	X	X	X	Previously Written Cursor			
	X	X	X	L	X	H	X													
	X	X	X	L	H	X	X													

L = LOGIC LOW INPUT
H = LOGIC HIGH INPUT
X = DON'T CARE

"a" = ASCII CODE CORRESPONDING TO SYMBOL "A"
NC = NO CHANGE
▩ = CURSOR CHARACTER (ALL SEGMENTS ON)

Figure 3a. HPDL-2416 Cursor/Data Memory Write Truth Table

Function	$\overline{BL}$	$\overline{CLR}$	CUE	$\overline{CU}$	$\overline{CE_1}$	$\overline{CE_2}$	$\overline{WR}$	DIG_3	DIG_2	DIG_1	DIG_0	
CUE	H	H	L	X	X	X	X	A	B	C	D	Display previously written data
	H	H	H	X	X	X	X	▩	▩	▩	▩	Display previously written cursor
Clear	H	L	X	X	X	X	X*	⬚	⬚	⬚	⬚	Clear data memory, cursor memory unchanged
	*NOTE: CLR should be held low for 4 ms following the last WRITE cycle to ensure all data is cleared.											
Blanking	L	X	X	X	X	X	X	⬚	⬚	⬚	⬚	Blank display, data and cursor memories unchanged.

Figure 3b. HPDL-2416 Displayed Data Truth Table

ELECTRICAL DESIGN CONSIDERATIONS

The on-board CMOS IC, incorporated in the HPDL-2416, simplifies the use of the display. The truth table is shown in Figure 3. Figure 3a shows how to write data to the ASCII and CURSOR memories. Figure 3b shows the operation of the $\overline{CURSOR\ SELECT}$, $\overline{BLANKING}$, AND $\overline{CLEAR}$ inputs. In addition to the truth table, correct timing is necessary for reliable operation of the part. Timing for the HPDL-2416 is in Figure 4. Individual examples show the correct logic levels to load and display data, load and use the cursor, and clear and blank the display. These examples are illustrated in Figures 5, 6, 7, 8 and 9.

Writing and Displaying ASCII Data

Figure 5 shows the appropriate logic levels to write and display ASCII characters. ASCII data is entered via the DATA inputs D_0-D_6 and is stored in the memory location indicated by the ADDRESS inputs A_0-A_1. The $\overline{CURSOR\ SELECT}$ should be logic 1 to write ASCII data to the display. As shown in the timing diagram of Figure 4, the $\overline{CHIP\ ENABLES}$, ADDRESS inputs and $\overline{CURSOR\ SELECT}$ input must be stable prior to the falling edge of the $\overline{WRITE}$ signal. The DATA inputs need to be stable prior to the rising edge of the $\overline{WRITE}$ signal. The DATA, ADDRESS, $\overline{CHIP\ SELECT}$ and $\overline{CURSOR\ SELECT}$ inputs must be held stable until a specified hold time following the rising edge of the $\overline{WRITE}$ signal.

ASCII data will be decoded and displayed as long as the logic levels shown in Figure 5b are maintained. It is possible to generate 128 different input codes with seven DATA inputs. Referring to the ASCII character set in Figure 1, all valid 64 character ASCII codes are created when DATA input D_5 is the logical complement of DATA input D_6. If DATA inputs D_5 and D_6 are in the same logical state, invalid data will be stored in the display memory. (Invalid data will be decoded by the IC as a blank character). Figure 6 shows the 6 bit 64 character ASCII character set. Designers who wish to use the 6 bit 64 character ASCII set can invert D_5 to generate D_6.

Parameter	Symbol	−20° C t_{MIN}	25° C t_{MIN}	70° C t_{MIN}	Units
Address Setup Time	t_{AS}	90	115	150	ns
Write Delay Time	t_{WD}	10	15	20	ns
Write Time	t_W	80	100	130	ns
Data Setup Time	t_{DS}	40	60	80	ns
Data Hold Time	t_{DH}	40	45	50	ns
Address Hold Time	t_{AH}	40	45	50	ns
Chip Enable Hold Time	t_{CEH}	40	45	50	ns
Chip Enable Setup Time	t_{CES}	90	115	150	ns
Clear Time	t_{CLR}	2.4	3.5	4.0	ms
Access Time		130	160	200	ns
Refresh Rate		420-790	310-630	270-550	Hz

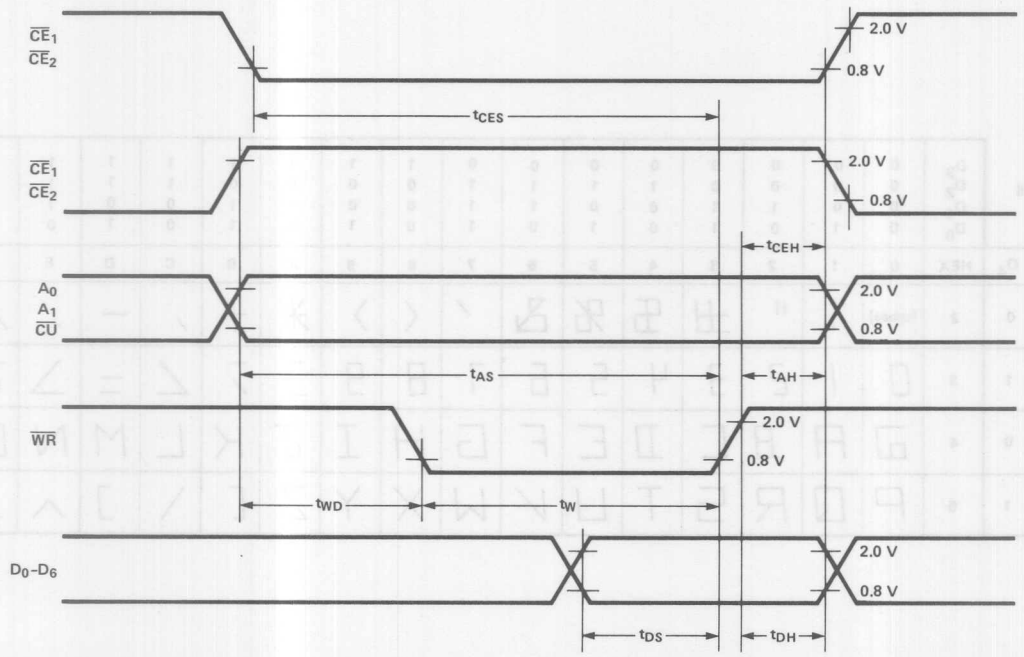


Figure 4. HPDL-2416 AC Timing Characteristics and Timing Diagram Guaranteed Over Operating Temperature

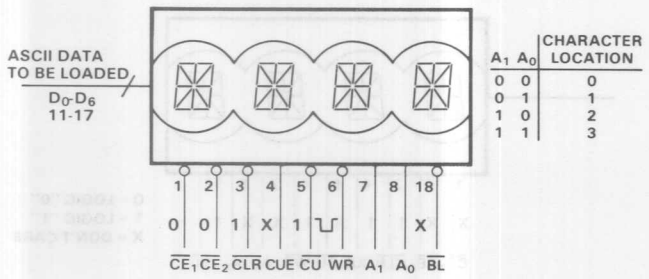


Figure 5a. Logic Levels to Load ASCII Memory

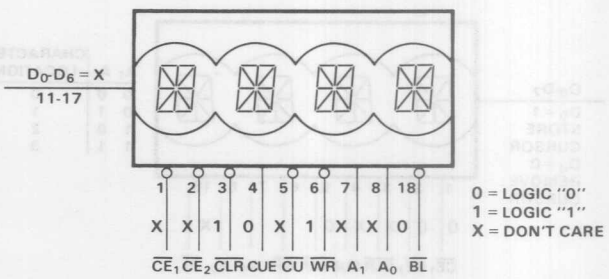


Figure 5b. Logic Levels to Display ASCII Memory

Writing and Displaying Cursor Data

Figure 7 illustrates the proper logic levels to write and display CURSOR data. The $\overline{\text{CURSOR SELECT}}$ should be logic 0 to write a cursor to the display. DATA input, D_0, is used by the IC to enter or remove a cursor from CURSOR memory. If $D_0 = 1$, a cursor will be entered; if $D_0 = 0$, a cursor will be removed. The ADDRESS inputs select the location in memory to store CURSOR data. Writing CURSOR data requires the same timing as writing ASCII data. As shown in the timing diagram of Figure 4 the $\overline{\text{CHIP ENABLES}}$, ADDRESS inputs and $\overline{\text{CURSOR SELECT}}$ input should be stable prior to the falling edge of the $\overline{\text{WRITE}}$ signal; DATA input, D_0, needs to be stable prior to the rising edge. D_0, ADDRESS, $\overline{\text{CHIP ENABLE}}$ and $\overline{\text{CURSOR SELECT}}$ inputs must be held stable until a specified hold time following the rising edge of the $\overline{\text{WRITE}}$ signal.

CURSOR data will be decoded and displayed as long as the logic levels shown in Figure 1b are maintained. If a location in CURSOR memory does not have a CURSOR stored in it, the CMOS IC will display the contents of the ASCII memory for that location. A flashing cursor can be generated by strobing the CURSOR ENABLE input.

Clear Function

Figure 8 shows the logic levels necessary to clear the display. The $\overline{\text{CLEAR}}$ input fills the ASCII memory with blanks without affecting the contents of the CURSOR memory. Clearing is accomplished by sequentially writing a blank character to each location in ASCII memory using the internal refresh clock. At worst case, the CMOS IC requires ms to clear the display. If the $\overline{\text{BLANKING}}$ input is logic 0, the $\overline{\text{CLEAR}}$ function will be inhibited.

BITS D_3 D_2 D_1 D_0		0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
D_5 D_4	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1 0	2	(space)	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
1 1	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0 0	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0 1	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_

Figure 6. HPDL-2416 6-bit ASCII Character Set

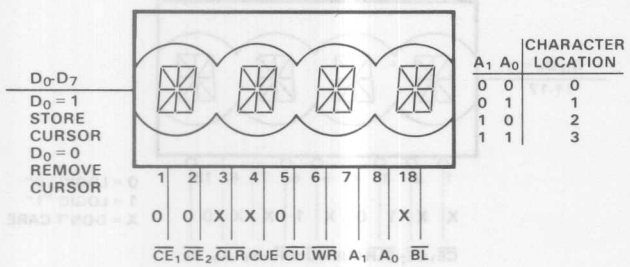


Figure 7a. Logic Levels to Load Cursor Memory

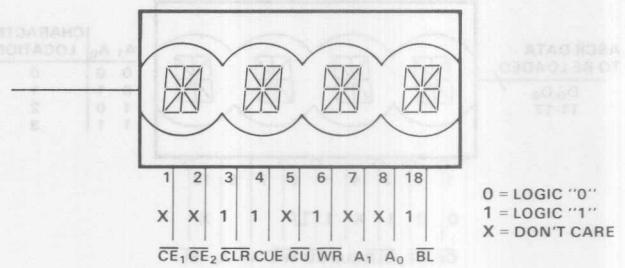


Figure 7b. Logic Levels to Display Cursor Memory

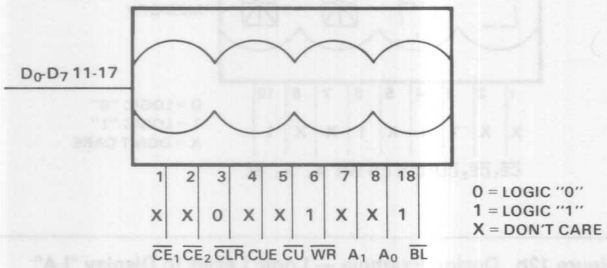


Figure 8. Logic Levels to Clear Display

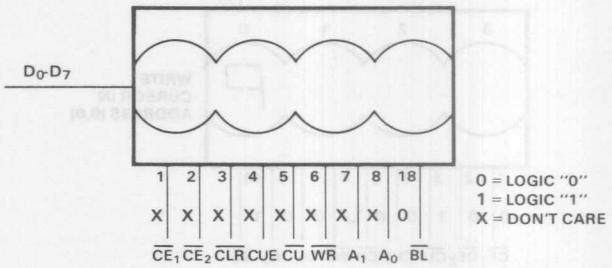


Figure 9. Logic Levels to Blank the Display

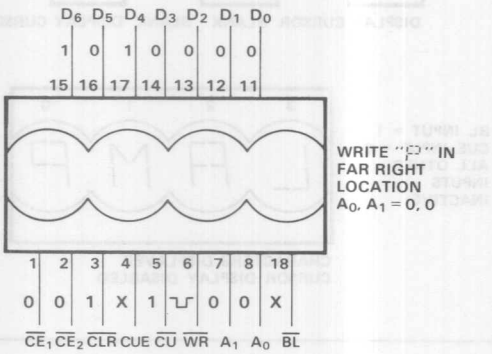


Figure 10a. Design Example — Logic Levels to Write "P"

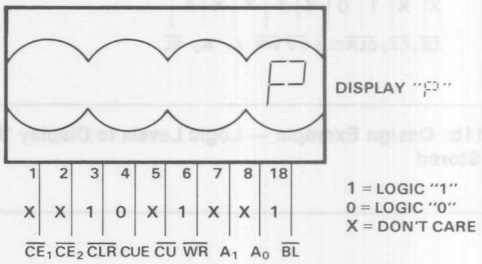


Figure 10b. Design Example — Logic Levels to Display "P"

Blanking Function

Figure 9 shows the logic levels needed to blank the HPDL-2416. The $\overline{\text{BLANKING}}$ input should be logic 1 for normal operation. Otherwise, if the $\overline{\text{BLANKING}}$ input is logic 0, the CMOS IC disables the segment drivers, the internal refresh counter and the $\overline{\text{CLEAR}}$ input causing the display to go blank. Neither ASCII memory nor CURSOR memory is affected by the $\overline{\text{BLANKING}}$ function. A flashing display can be created by strobing the $\overline{\text{BLANKING}}$ input. Since the $\overline{\text{BLANKING}}$ input disables the refresh counter, a strobing frequency faster than 5 Hz may interfere with the internal refreshing of the display.

Design Example

The following example illustrates how the HPDL-2416 functions. The design objective is to display a message that consists of the word L A M P plus a flashing cursor in the two far right locations.

1. Loading Data

Figure 10 illustrates how the letter P is loaded and displayed. Characters can be entered randomly without need for synchronizing the external WRITE signal to the internal clock. Letters L, A and M are loaded in the same manner as the letter P; the only changes consist of using the correct ASCII code and address.

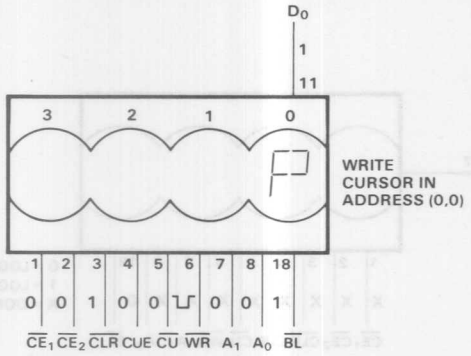


Figure 11a. Design Example — Logic Levels to Write Cursor in the "P" Location

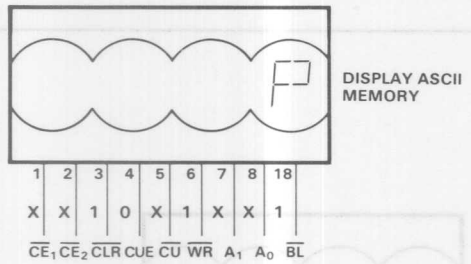


Figure 11b. Design Example — Logic Levels to Display "P" with Cursor Stored

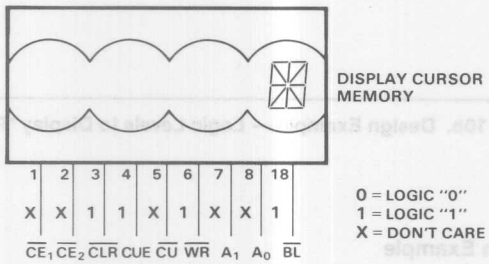


Figure 11c. Design Example — Logic Levels to Display Cursor

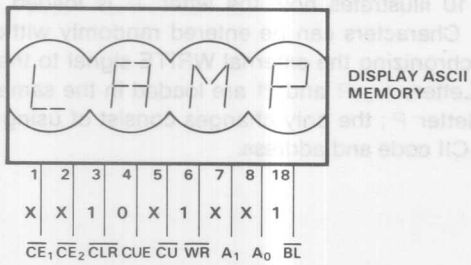


Figure 12a. Design Example — Logic Levels to Display "LAMP"

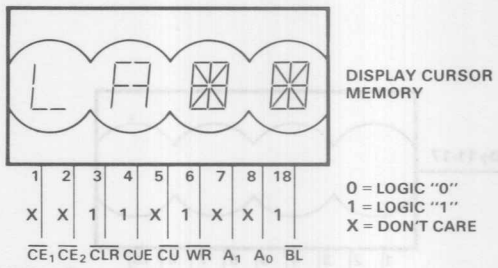


Figure 12b. Design Example — Logic Levels to Display "LA"

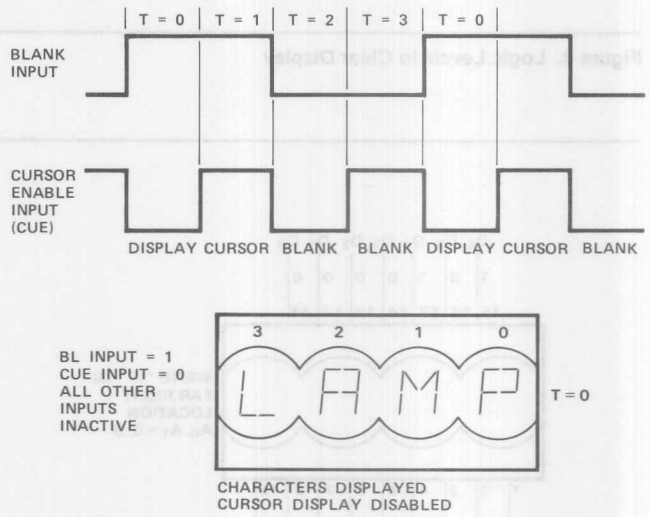


Figure 13a. Design Example — Logic Levels to Display "LAMP"

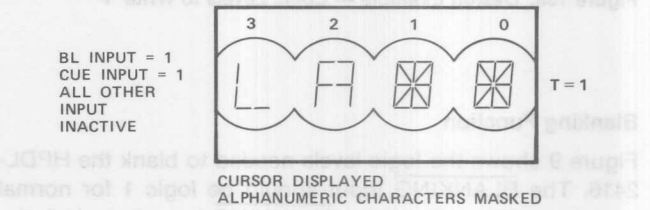


Figure 13b. Design Example — Logic Levels to Display "LA"

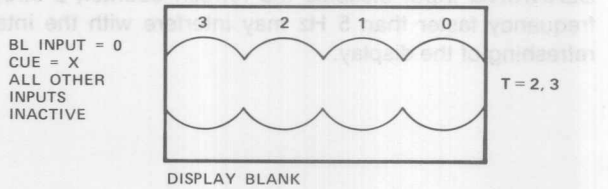


Figure 13c. Design Example — Logic Levels to Blank Display

2. Loading Cursor

Figure 11 illustrates how a CURSOR is loaded into the display using the same address as the letter P. (It is important to note letter P is still being displayed). CURSORS may be written in or removed from any character location by using the DATA input, D_0, and the corresponding addresses.

3. Flashing Cursor and Blanking

As shown in Figure 12, strobing the CUE input causes the HPDL-2416 to alternate between displaying L A M P and displaying two cursors. Exceeding the strobing limit of 5 Hz may cause the eye to perceive only the cursor. If $\overline{\text{BLANKING}}$ = 0, the display will be blank. Strobing both CURSOR and $\overline{\text{BLANKING}}$ with the waveforms shown in Figure 13 will cause the HPDL-2416 ENABLE to display the following sequence: L A M P, L A ▩ ▩, BLANK, BLANK, L A M P ...

USING THE HPDL-2416 WITH MICROPROCESSORS

Several different techniques can be used to interface an HPDL-2416 to a microprocessor. The HPDL-2416 is sufficiently versatile to be configured as either a memory device in a memory mapped system or as an output device in an I/O mapped system. Both types of systems will be covered. All microprocessor addresses in the following examples have been selected for illustrative purposes only. They can be changed to suit the designer's needs.

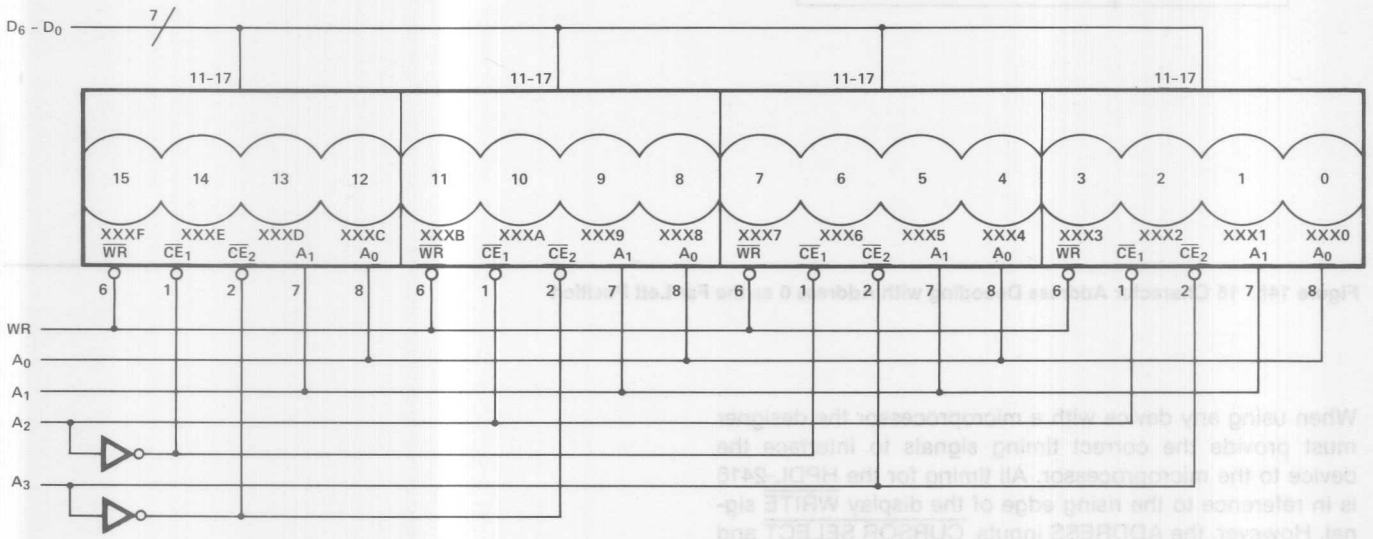


Figure 14a. 16 Character Address Decoding with Address 0 as the Far Right Position

MEMORY MAPPED SYSTEM

The easiest method is to configure the HPDL-2416 as a memory device. Each character in the display will use one memory location. A 16 character length display requires the use of 16 memory locations. Address decoding is simplified by using the two $\overline{\text{CHIP ENABLE}}$s. For a 16 character display, microprocessor address lines A_0 and A_1 would be connected directly to four HPDL-2416 displays. Using two inverters to generate A_2 and A_3, the appropriate combinations of A_2, A_2, A_3 and A_3 would be connected to the $\overline{\text{CHIP ENABLE}}$s to select the appropriate display. Thus, the 16 character display would be located at microprocesor addresses XXX0 to XXXF. Figure 14 shows how a 16 character display system can be formatted to allow for the 16 addresses to be organized with either address 0 or address F being the far left display character.

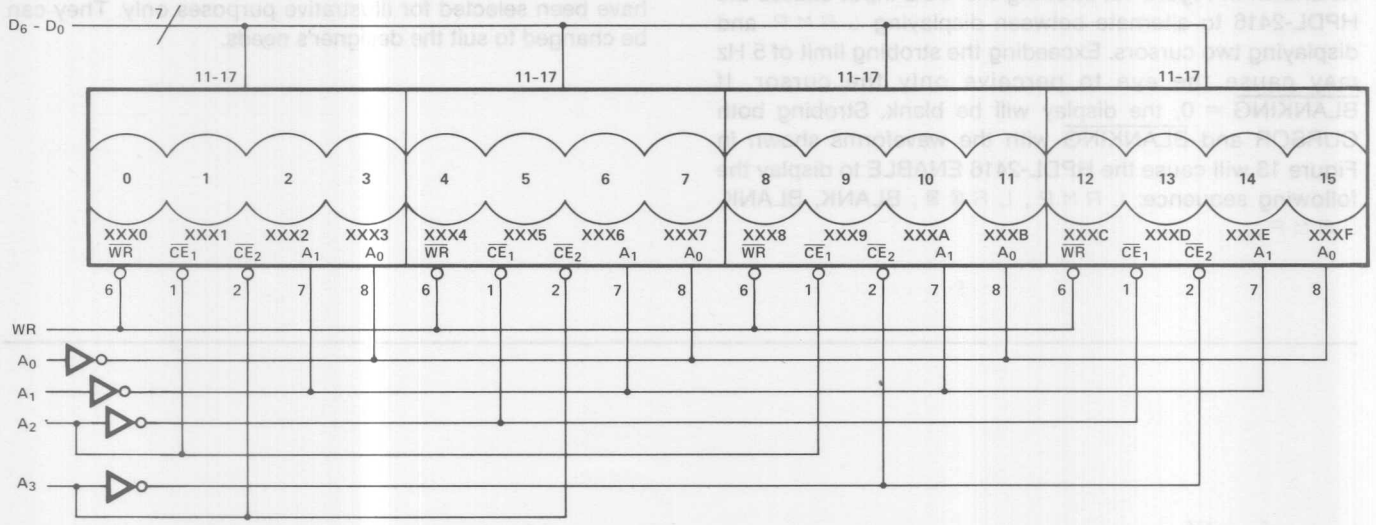


Figure 14b. 16 Character Address Decoding with Address 0 as the Far Left Position

When using any device with a microprocessor the designer must provide the correct timing signals to interface the device to the microprocessor. All timing for the HPDL-2416 is in reference to the rising edge of the display $\overline{\text{WRITE}}$ signal. However, the ADDRESS inputs, $\overline{\text{CURSOR SELECT}}$ and $\overline{\text{CHIP ENABLES}}$ must be stable prior to the falling edge. Figure 15 compares the timing of the HPDL-2416 to the Motorola 6800, Intel 8085 and Zilog Z-80.

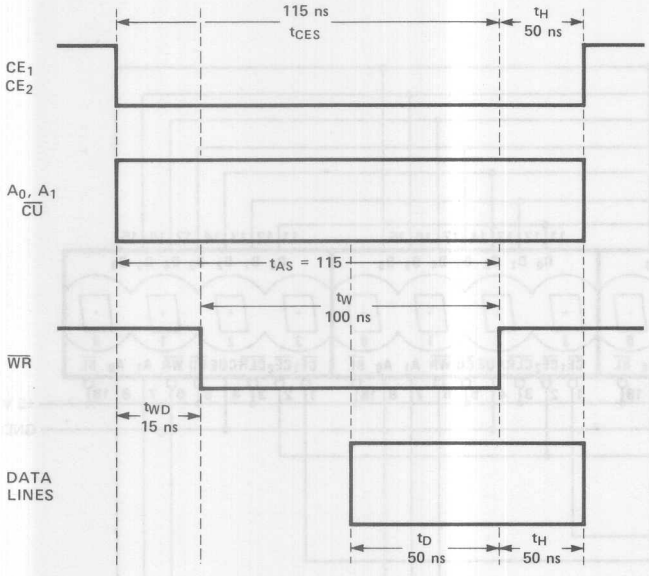
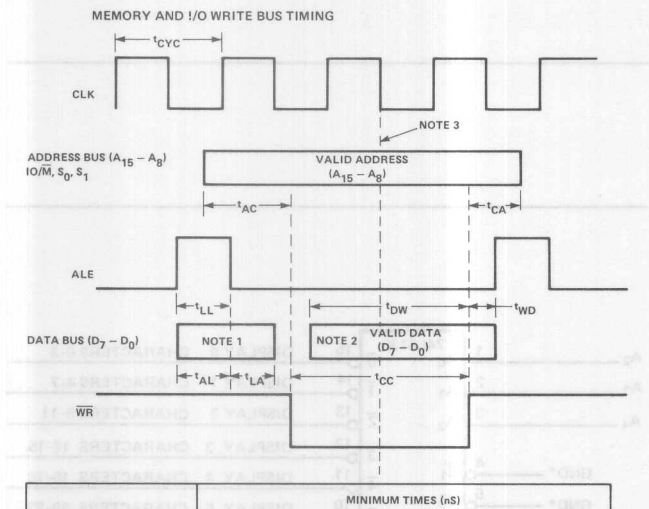


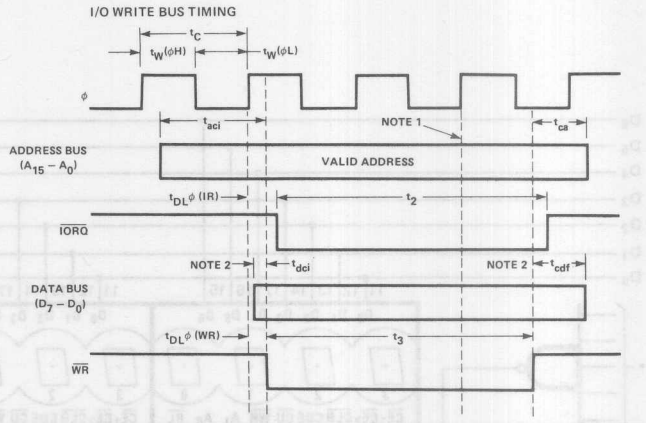
Figure 15a. HPDL-2416 Write Timing



8085 MICROPROCESSOR	MINIMUM TIMES (nS)							
	t_{LL}	t_{AL}	t_{LA}	t_{AC}	t_{CC}	t_{DW}	t_{WD}	t_{CA}
8085A, t_{CYC} = 320	140	110	100	270	400	420	100	120
8085A–2, t_{CYC} = 200	80	50	50	115	230	230	40	60

NOTE 1: VALID ADDRESS ($A_7 - A_0$). User should externally latch this data on ↓ edge of ALE.
NOTE 2: INTEL does not guarantee that data is valid prior to ↓ edge of $\overline{WR}$ (see Note 5, pg. 2-15 of 8085 USERS MANUAL, June 1977).
NOTE 3: Additional wait cycles can be inserted here.
From INTEL Component Data Catalog, 1978

Figure 15b. Memory and I/O Write Timing for the Intel 8085A Microprocessor Family

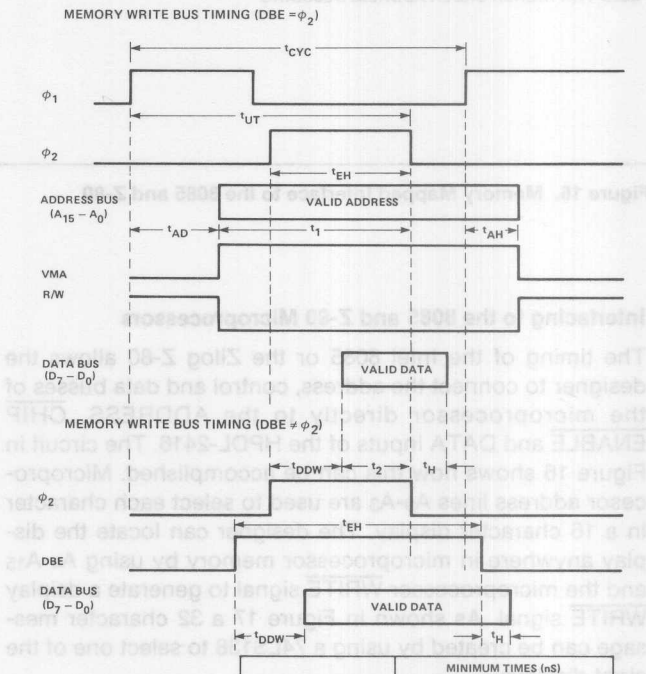


Z-80 MICROPROCESSOR	MINIMUM TIMES (nS)					
	t_{aci}	t_{dci}	t_3	t_2	t_{ca}	t_{cdf}
Z-80, t_c = 400	320	-30	900	890	140	100
Z-80A, t_c = 250	180	-60	545	535	60	40

$t_{aci} = t_c - [80\ (Z\text{-}80), 70\ (Z\text{-}80A)]$
$t_{dci} = t_W(\phi L) - [210\ (Z\text{-}80), -170\ (Z\text{-}80)]$
$t_3 = 2t_c + t_W(\phi H) - t_{DL\phi}$ (WR)
$t_2 = 2t_c + t_W(\phi H) - t_{DL\phi}$ (IR)

NOTE 1: Additional wait cycles can be inserted here.
NOTE 2: Zilog does not guarantee the relative timing between $\overline{IORQ}$ and $\overline{WR}$.
From ZILOG Z-80 CPU Product Specification, March, 1978

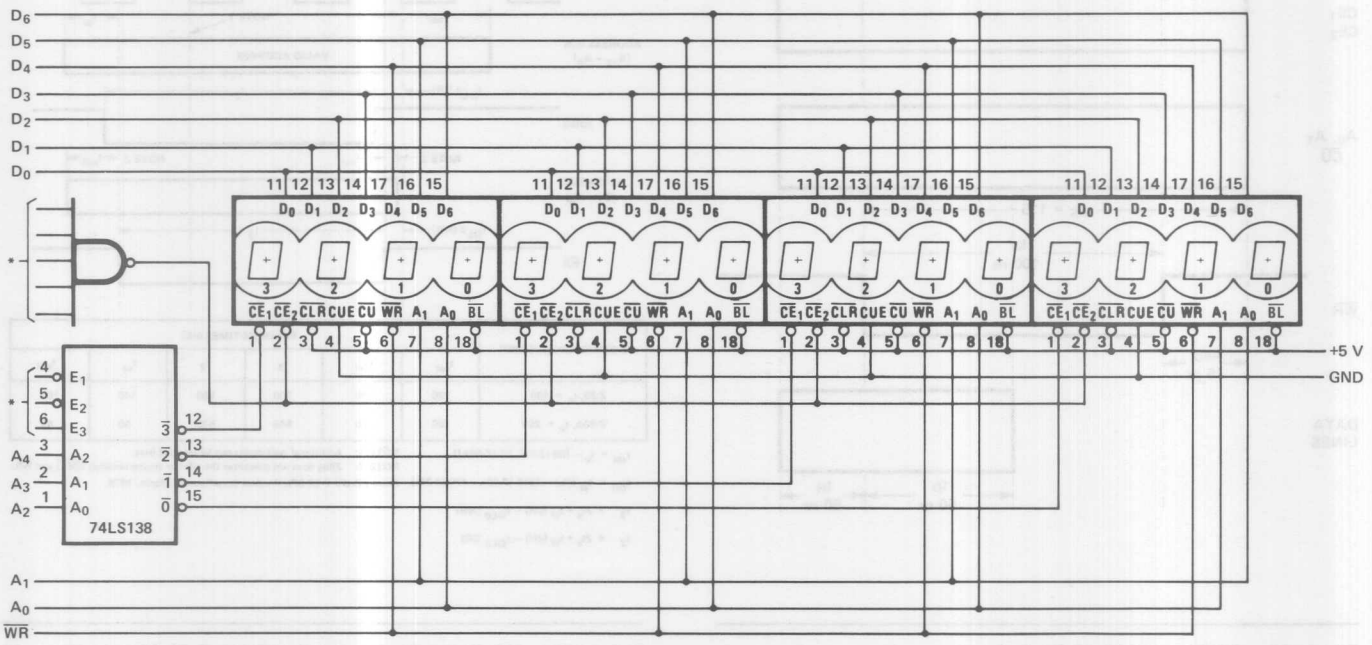
Figure 15c. I/O Write Timing for the Zilog Z-80 Microprocessor Family



6800 MICROPROCESSOR	MINIMUM TIMES (nS)				
	t_{DDW} (MAX)	t_1	t_{AH}	t_2	t_H
6800, t_{CYC} = 1000	225	630	30	225	10
68A00, t_{CYC} = 666	200	420	30	80	10
68B00, t_{CYC} = 500	160	290	30	60	10

$t_1(MIN) = t_{UT}(MIN) - t_{AD}(MAX)$
$t_2(MIN) = t_{EH}(MIN) - t_{DDW}(MAX)$
From MOTOROLA SEMICONDUCTOR MC6800 Data Sheet (DS9471), 1978

Figure 15d. Memory Write Timing for the Motorola 6800 Microprocessor Family



*USED FOR HIGHER ORDER ADDRESS DECODING

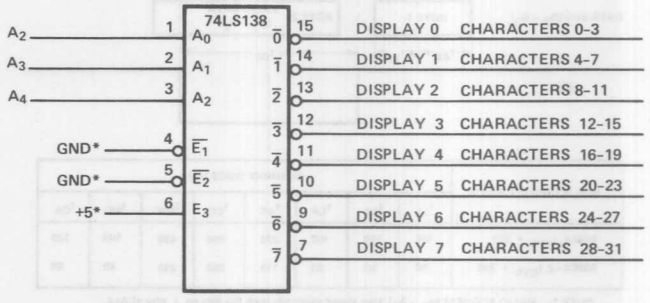
Figure 16. Memory Mapped Interface to the 8085 and Z-80

Interfacing to the 8085 and Z-80 Microprocessors

The timing of the Intel 8085 or the Zilog Z-80 allows the designer to connect the address, control and data busses of the microprocessor directly to the ADDRESS, $\overline{\text{CHIP ENABLE}}$ and DATA inputs of the HPDL-2416. The circuit in Figure 16 shows how this can be accomplished. Microprocesor address lines A_0-A_3 are used to select each character in a 16 character display. The designer can locate the display anywhere in microprocessor memory by using A_4-A_{15} and the microprocessor $\overline{\text{WRITE}}$ signal to generate a dsiplay $\overline{\text{WRITE}}$ signal. As shown in Figure 17 a 32 character message can be created by using a 74LS138 to select one of the eight displays.

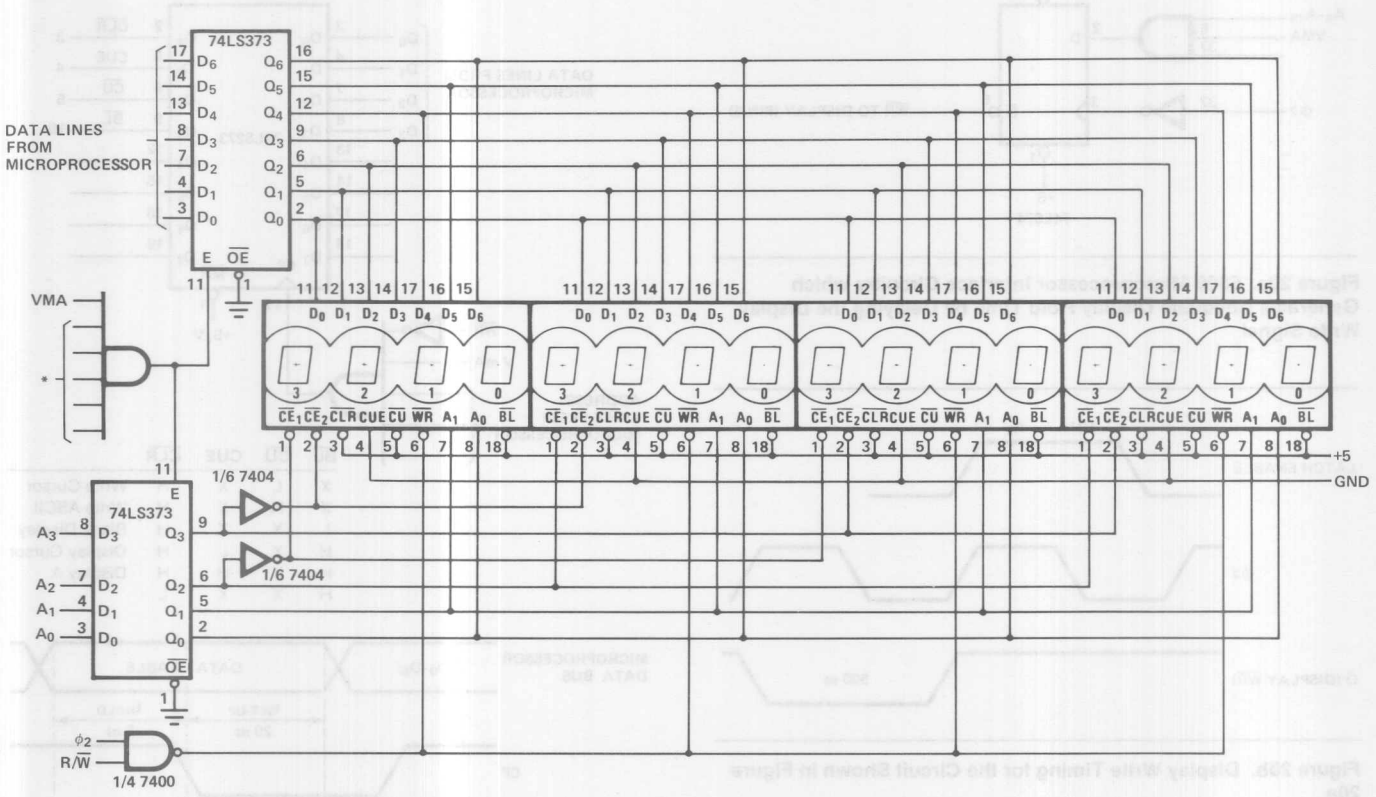
Interfacing to the 6800

Two transparent latches, shown in Figure 18, can be used to provide the necessary hold time for the HPDL-2416 when interfacing to a Motorola 6800. Both latches are enabled on the rising edge of the ENABLE signal. Information is latched on the falling edge of the ENABLE signal. Data is stored in one latch and address information is stored in the other. The address of the display in microprocessor memory is defined by a logical combination of A_4-A_{15} and VMA which is used to generate an ENABLE signal for these latches.



*USED FOR HIGHER ADDRESS DECODING ORDER.

Figure 17. 32 Character Address Decoder Technique for the HPDL-2416



*USED FOR HIGHER ORDER ADDRESS DECODING.

Figure 18. Memory Mapped Interface for the 6800

It is essential that the display $\overline{\text{WRITE}}$ signal generated by the interface circuitry allows the DATA inputs to stabilize prior to the rising edge. Figures 19 and 20 illustrate two circuits can be used to provide the necessary set-up time. The first technique, shown in Figure 19, takes advantage of the independence of the display $\overline{\text{WRITE}}$ cycle and the internal refresh cycle. In this case, data is written to the display on every microprocessor READ/$\overline{\text{WRITE}}$ cycle. A display $\overline{\text{WRITE}}$ signal is created by Nanding 0_2 with the microprocessor $\overline{\text{WRITE}}$ signal. Even though the display is being written during every microprocessor READ/$\overline{\text{WRITE}}$ cycle, the display message is unaffected since the data in the latches remains unchanged.

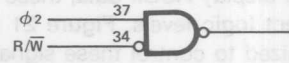


Figure 19a. 6800 Microprocessor Interface Circuitry which Generates Adequate Display Hold Time by Multiple Display Writes

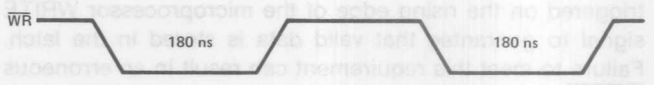


Figure 19b. Display Write Timing for the Circuit Shown in Figure 19a

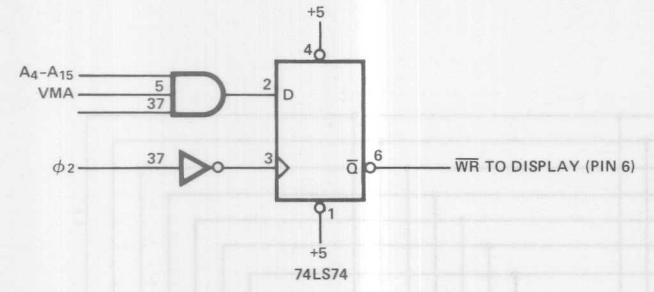


Figure 20a. 6800 Microprocessor Interface Circuitry which Generates Adequate Display Hold Time by Delaying the Display Write Signal

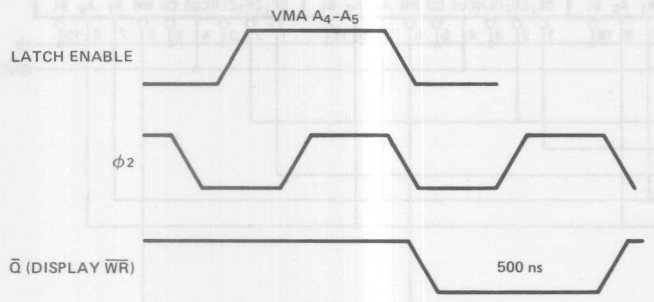


Figure 20b. Display Write Timing for the Circuit Shown in Figure 20a

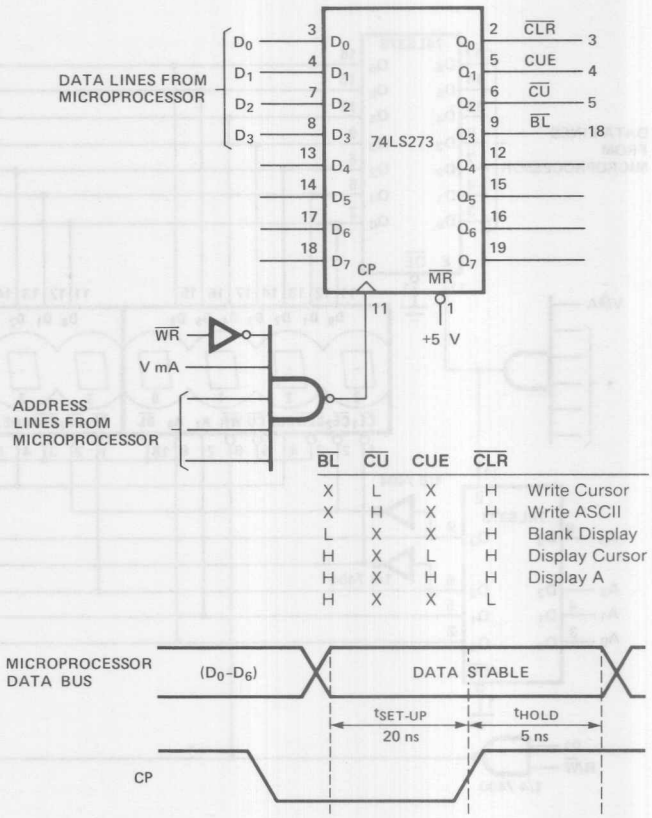


Figure 21. Use of 74LS273 to Store Control Signals for the HPDL-2416

The other technique, illustrated in Figure 20, is to delay the display $\overline{\text{WRITE}}$ pulse. This is accomplished by using 0_2 to clock a D-type flip-flop. A_4-A_{15} and VMA are connected to the D-input of the flip-flop. The $\overline{Q}$ output of the flip-flop is used as the display $\overline{\text{WRITE}}$.

Using the Cursor, Blanking and Clear Functions

The CURSOR, $\overline{\text{BLANKING}}$ and $\overline{\text{CLEAR}}$ functions control the display output. To effectively use these display functions external storage of this information is required. However, if the designer only wants to display ASCII data, these inputs can be tied to their correct logic levels. Figure 21 shows how a latch can be organized to control these signals and the timing of the clock to the latch. The microprocessor address lines and $\overline{\text{WRITE}}$ signal can be used to locate the latch in microprocessor memory and to clock the latch.

Before using the display, the latch needs to be loaded with the appropriate control word to enter data. After data has been written into the display, the designer can control the display output by changing the contents of the latch. To ensure reliable operation of the display an EDGE TRIGGERED LATCH MUST BE USED. The latch should be triggered on the rising edge of the microprocessor $\overline{\text{WRITE}}$ signal to guarantee that valid data is stored in the latch. Failure to meet this requirement can result in an erroneous $\overline{\text{CLEAR}}$ signal.

I/O MAPPED SYSTEM

The I/O mapped system uses a peripheral interface adaptor (PIA). One method of using a PIA is to configure it as a latch. An alternate technique is to take advantage of the PIA's handshaking capabilities to provide a display $\overline{\text{WRITE}}$ signal. Both configurations will be discussed.

Latch Technique

The schematic in Figure 22 shows how an 8255 PIA can be used to interface an HPDL-2416 to an 8085 microprocessor. Three output ports from the 8255 are used to write data to the HPDL-2416. Port A is used to store data, port B is used to store display function information and port C is used to store display address data.

Figure 23 lists four subroutines needed to control the display. Before the display can be used, the PIA has to be initialized. The INIT routine formats the PIA as three 8 bit output ports. Both the CLCUR and the TEXT subroutines use the $\overline{\text{WRITE}}$ subroutine to generate a display $\overline{\text{WRITE}}$ signal. A display $\overline{\text{WRITE}}$ is created by the software switching C_7 from a logic high to a logic low and back to a logic high without changing any other data. The CLCUR will remove cursors from each location in the display. The text subroutine will write a 16 character message from a table located in the microprocessor. Once the PIA has been done by changing the contents of the message table at microprocessor addresses 0900 to 090F or by changing the pointer, TABLE+10H, in the text subroutine to a new message location. The designer can control the display output by changing the contents of port B after data has been written into the display.

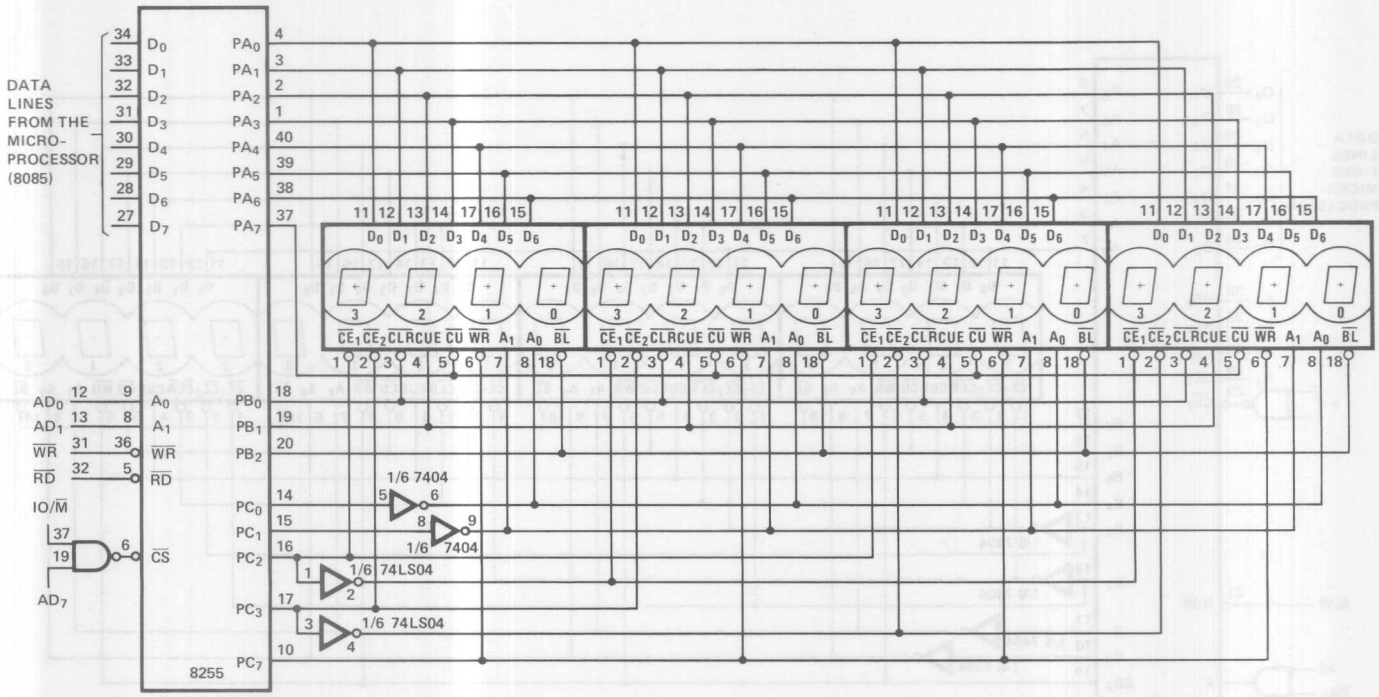


Figure 22. 8255 PIA Interface to the HPDL-2416

```
      OBJECT
LOC   CODE      SOURCE STATEMENTS
00    00        PortA    EQU  00H
01    01        PortB    EQU  01H
02    02        PortC    EQU  02H
03    03        Init     EQU  03H
0900  0900      Table    EQU  0900H
0800  0800      Initial  EQU  0800H
0810  0810      Write    EQU  0810H
0820  0820      Clcur    EQU  0820H
0840  0840      Text     EQU  0840H

0800  3E 80     INITIAL  MVI  A,80H      Initialize PIA as Three Output Ports
0802  D3 03              OUT  INIT
0804  3E 05              MVI  A,05H
0806  D3 01              OUT  PortC
0808  C9                 RET

0810  F6 80     WRITE    ORI  A,80H      Load Address and Set WRITE Pin High.
0812  D3 02              OUT  PortC
0814  E6 7F              ANI  A,7FH      Set WRITE pin low.
0816  D3 02              OUT  PortC
0818  F6 80              ORI  A,80H      Set WRITE Pin High.
081A  D3 02              OUT  PortC      Write Cycle Complete.
081C  C9                 RET             Return.

0820  3E 00     CLCUR    MVI  A,00H      Select Cursor Memory and Set D0 to 0.
0822  D3 00              OUT  PortA
0824  D6 0F              MVI  B,0FH      Set Length of Message.
0826  78         CLC      MOV  A,B        Move Display Address To Acc.
0827  CD 1008            CALL WRITE      Call WRITE Subroutine.
082A  05                 DCR  B          Decrement For Next Address
082B  C2 2608            JNZ  CLC        Loc 0?, No Jump CLC, Yes Continue.
082E  78                 MOV  A,B        Clear Last Cursor Location.
082F  CD 1008            CALL WRITE
0832  C9                 RET             Return.

0840  3E 05     TEXT     MVI  A,05H      Set Display To Show ASCII
0842  D3 01              OUT  PortB
0844  06 00              MVI  B,00H      Set Pointer To Right Most Character Loc

0846  21 1009            LXI  H,TABLE+10H
0849  7E         NEXT     MOV  A,M        Load ASCII Data In PIA.
084A  D3 00              OUT  PortA
084C  79                 MOV  A,B        Move Character Address To Acc.
084D  CD 1008            CALL WRITE      Call WRITE Subroutine
0850  2B                 DCX  H          Decrement Pointer
0851  04                 INR  B          Increment To Next Display Address.
0852  3E 10              MVI  A,10H
0854  B8                 CMP  B          End Of Message, Yes Continue;
0855  C2 4808            JNZ  NEXT       No, Jump To NEXT.
0858  3E 05              MVI  A,05H
085A  D3 01              OUT  PortB      Set Display Control.
085C  C9                 RET             Return.
```

Figure 23. 8085 Microprocessor Program to Interface to the Circuit in Figure 22

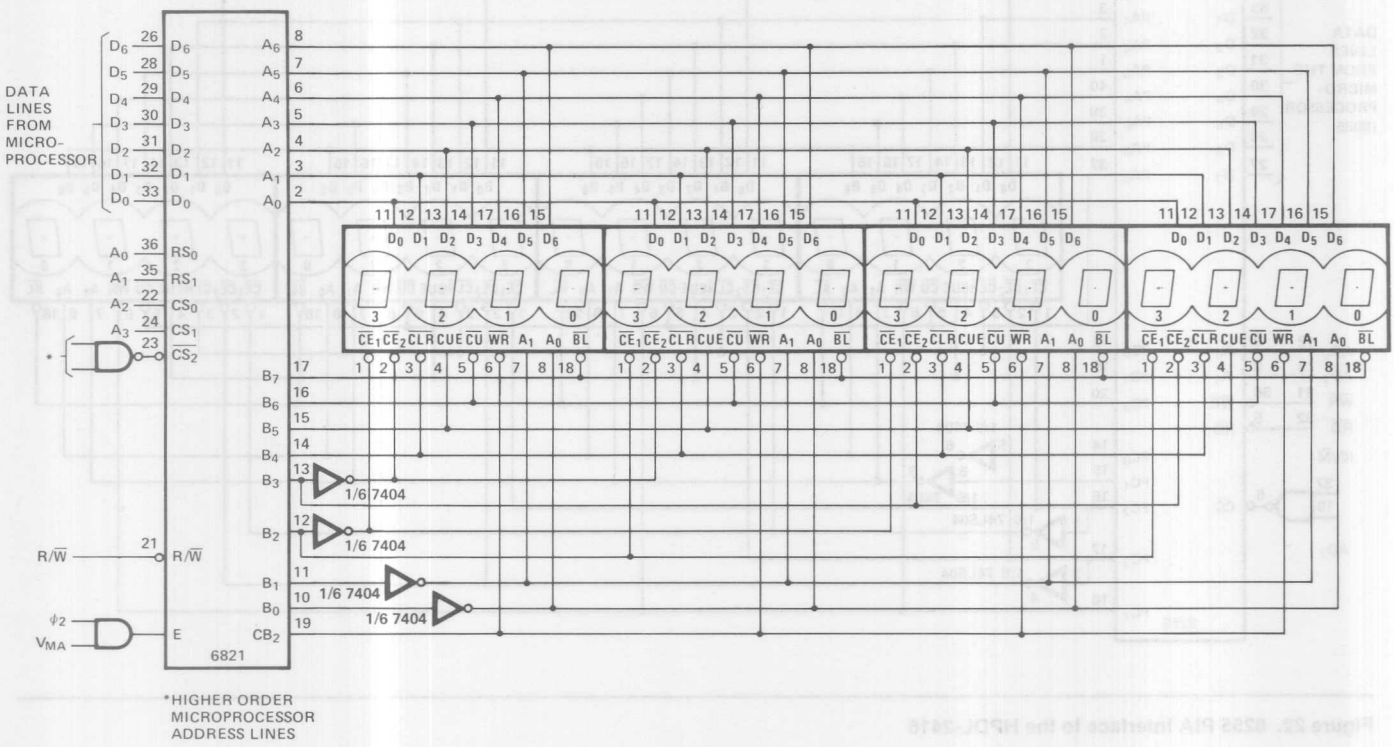


Figure 24. 6821 PIA to Interface to the HPDL-2416

Location	Object Code	Source Statements		
	0F 31	Control A	EQU	$ 0F 31
	0F 30	DDA	EQU	$ 0F 30
	0F 30	PortA	EQU	$ 0F 30
	0F 33	Control B	EQU	$ 0F 33
	0F 32	DDB	EQU	$ 0F 32
	0F 32	Port B	EQU	$ 0F 32
	05 00	Init	EQU	$ 0500
			ORG	$ 0500
0500	86 00	Init	LDA A	I, $00
0502	C6 FF		LDA B	I, $FF
0504	B7 0F 31		STA A	E, Control A
0507	F7 0F 30		STA B	E, DDA
050A	B7 0F 33		STA A	E, Control B
050D	F7 0F 32		STA B	E, DDB
0510	86 2C		LDA A	I, $2C
0512	B7 0F 31		STA A	E, Control A
0515	B7 0F 33		STA A	E, Control B
0518	39		RST	

Figure 25. Program to Initialize a 6821 PIA.

Handshaking Technique

The circuit in Figure 24 is designed to use a 6821 PIA as an interface between a 6800 microprocessor and an HPDL-2416. The PIA requires a minimum of four microprocessor address locations, with two addresses for each port. The PIA has to be initialized before it can be used. Figure 25 shows how the PIA must be initialized to operate the circuit in Figure 24. Ports A and B are set up as outputs.

The 6821 PIA is configured to generate a display $\overline{\text{WRITE}}$ signal every time display control information is written into port B. This eliminates the need for a $\overline{\text{WRITE}}$ subroutine. Ports A and B have data bits assigned as shown in Figure 26. Table 1 shows how the output of port B controls the display system. Figure 27 shows the timing between the PIA enable and CB2 (display $\overline{\text{WRITE}}$).

Control Register Configuration for Port A

Microprocessor Data Bus Configuration								
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0	6821 Port A Control Register Designation
X	X	X	X	X	1	X	X	Port A is selected
X	X	X	X	X	0	X	X	Data direction register of Port A is selected

Control Register Configuration for Port B

Microprocessor Data Bus Configuration								
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0	6821 Port B Control Register Designation
X	X	1	0	1	1	X	X	Port B selected to provide a write pulse to the display.
X	X	X	X	X	0	X	X	Data direction register B is selected.

6821 A and B Port Bit Assignment

Microprocessor Data Bus Configuration								
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Port A Data Register Designation
X	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Port A Display Data

Microprocessor Data Bus Configuration								
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	Port B Data Register Designation
$\overline{BL}$	$\overline{CU}$	$\overline{CUE}$	$\overline{CLR}$	$\overline{CE_2}$	$\overline{CE_1}$	A_1	A_0	Port B Display Control

Figure 26. Configuration of the 6821 PIA

Table 1.

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0	
$\overline{BL}$	$\overline{CU}$	$\overline{CUE}$	$\overline{CLR}$	$\overline{CE_2}$	$\overline{CE_1}$	A_1	A_0	Port B Bits
0	0	0	0	X	X	X	X	Blank display & write cursor
0	0	0	1	X	X	X	X	Blank display & write cursor
0	0	1	0	X	X	X	X	Blank display & write cursor
0	0	1	1	X	X	X	X	Blank display & write cursor
0	1	0	0	X	X	X	X	Blank display & write ASCII data
0	1	0	1	X	X	X	X	Blank display & write ASCII data
0	1	1	0	X	X	X	X	Blank display & write ASCII data
1	0	0	0	X	X	X	X	Use only to clear display
1	0	0	1	X	X	X	X	Write cursor & display ASCII data
1	0	1	0	X	X	X	X	Use only to clear display
1	0	1	1	X	X	X	X	Write cursor & display cursor
1	1	0	0	X	X	X	X	Use only to clear display
1	1	0	1	X	X	X	X	Write ASCII data & display ASCII data
1	1	1	0	X	X	X	X	Use only to clear display
1	1	1	1	X	X	X	X	Write ASCII data & display cursor

X = Do Not Care
1 = Logic High
0 = Logic Low

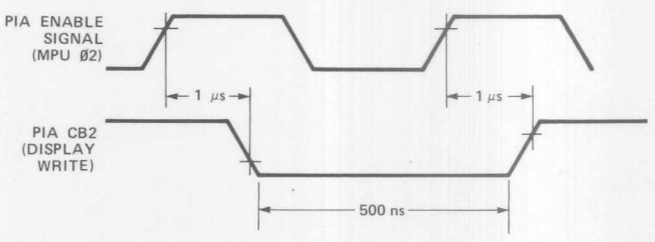


Figure 27. Display Write Timing for the 6821 PIA Circuit Shown in Figure 24

Location	Object Code	Source Statements	
	0F 30	Port A	EQU $ 0F30
	0F 32	Port B	EQU $ 0F32
	06 00	Table	EQU $ 0600
			ORG $ 0550
0550	C6 DF		LDA B I, $DF
0552	CE 0600		LDX Table
0555	A6 00	NCHAR	LDA A X, $00
0557	B7 0F 30		STA A E, Port A
055A	F7 0F 32		STA B E, Port B
055D	5A		DEC B
055E	08		INX
055F	C1 CF		CMP B I, $CF
0561	26 F2		BNE NCHAR
0563	39		RTS

Figure 28. 6800 Microprocessor Subroutine to Write a 16 Character Message via a 6821 PIA

Location	Hex Data	Character	
0600	42	B	Left Most Character
0601	55	U	
0602	59	Y	
0603	20	*Space*	
0604	48	H	
0605	50	P	
0606	20	*Space*	
0607	44	D	
0608	49	I	
0609	53	S	
060A	50	P	
060B	46	L	
060C	41	A	
060D	59	Y	
060E	49	S	
060F	20	*Space*	Right Most Character

Figure 29. Message Table for 6800 Microprocessor Subroutine Listed in Figure 28

Figure 28 lists a subroutine capable of operating a 16 character display. The subroutine works as follows:

Figure 29 shows how to organize a message table for the subroutine listed in Figure 28.

PREPROGRAMMED MESSAGE

There are many occasions when a designer needs to use an alphanumeric display. In some applications, the designer simply wishes to display one of several fixed status messages. While a microprocessor can be dedicated to this application, microprocessor time and memory are required. Microprocessor interaction can be minimized by using a display interface which includes all circuitry needed for text generation. Then, the microprocessor simply needs to select the desired message through a binary weighted input. This type of display interface is called a preprogrammed message system. The preprogrammed message system appears to the microprocessor as a very fast write only memory which uses only one memory location.

The circuit in Figure 30 is designed as a preprogrammed message system. The system shown is capable of displaying 128 different character messages that are 16 characters in length. The system can be modified to operate with different character lengths by varying the length of the counters and ROM size. The microprocessor selects a preprogrammed message to be displayed by using a binary weighted code. The system then feeds the message to be displayed to the HPDL-2416 one character at a time.

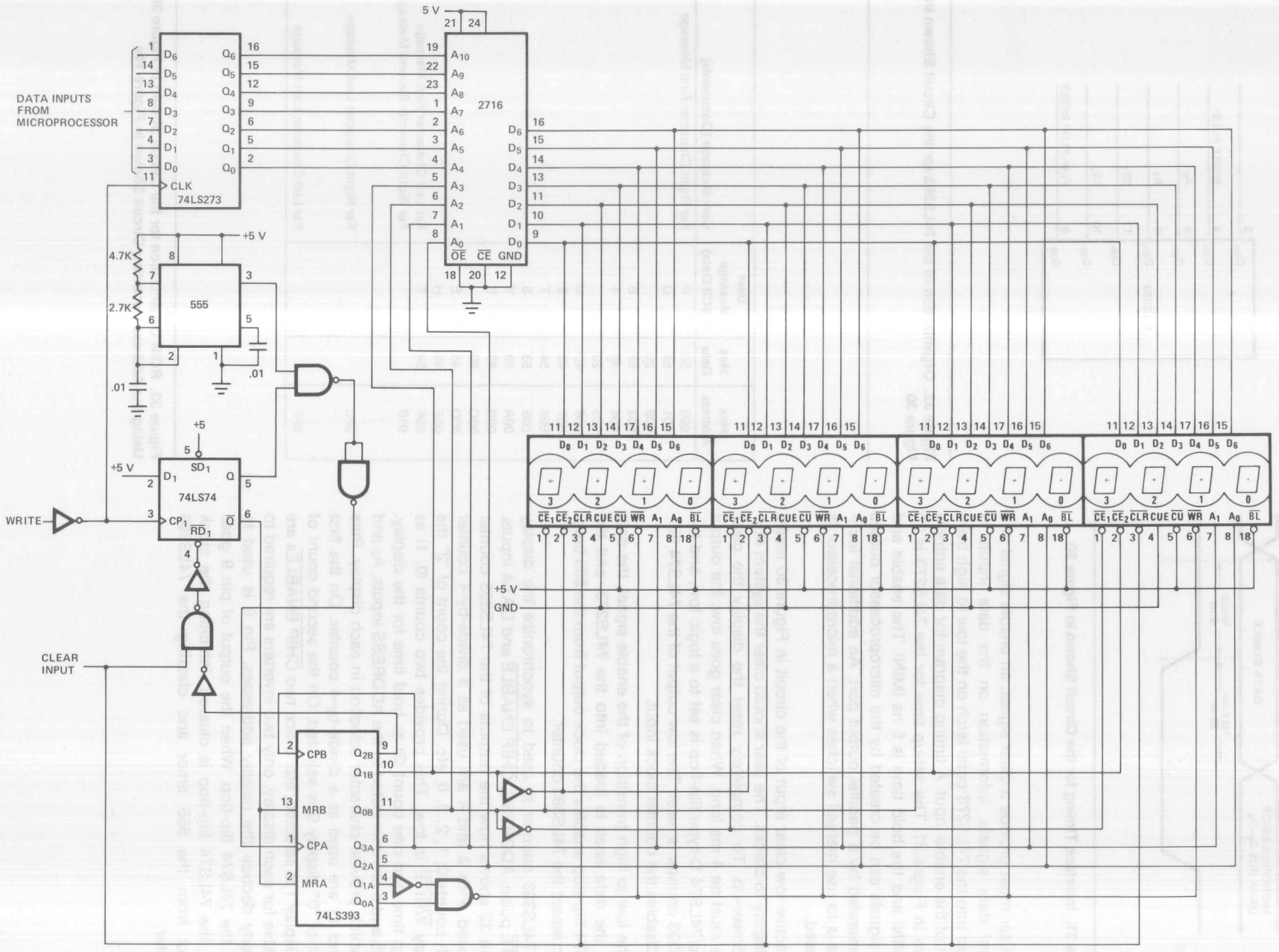


Figure 30. Preprogrammed Message System

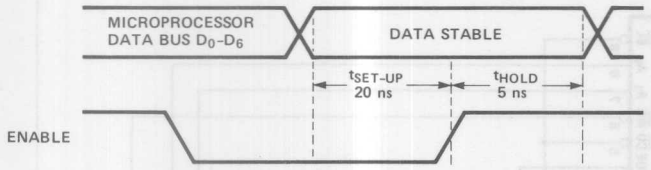


Figure 31. Interface Timing for the Circuit Shown in Figure 30

The user must provide a clear signal, an enable signal and several data signals. Information on the data inputs is loaded into the 74LS273 octal latch on the low to high transition of the enable input. A timing diagram for data entry is shown in Fiugre 31. The set-up time for the 74LS273 is 20 ns (MIN) and the hold time is 5 ns (MIN). The enable and data signals can be created by the microprocessor busses or generated by a parallel output port. An additional technique is to use manual switches when a microprocessor is not used.

The active low clear input of the circuit in Figure 30 resets the display to blanks. The user should clear the system during power-up. To completely reset the display the clear pulse must be 4 ms long. When clear goes low, the output of the 74LS74 D-type flip-flop is set to a logic low, and the 74LS393 counter is reset. The low output of the 74LS74 flip-flop disables the counter clock input.

On the low to high transition of the enable signal, the data from the data lines is loaded into the 74LS273, and the 74LS74 flip-flop enables the clock output from the 555 timer to increment the 74LS393 counter.

The 74LS393 counter is used to synchronize the display $\overline{\text{WRITE}}$ pulse, ADDRESS, $\overline{\text{CHIP ENABLE}}$ and DATA inputs. Figure 32 shows how the outputs of the 74LS393 counter are used. Pins 3 and 4 are used as a divide-by-4 counter which counts 0, 1, 2, 3, 0, etc. During the count of "2" the display $\overline{\text{WRITE}}$ is low. This provides two counts (0, 1) as set-up time and one count (3) as hold time for the display. Pins 5 and 6 are connected to the ADDRESS inputs, A_0 and A_1, which provide character selection in each display. Pins 10 and 11 are used as a divide-by-4 counter. On the first count of "0" display 0 is selected. On the second count of "1" display 1 is selected, etc. Since two $\overline{\text{CHIP ENABLE}}$s are available for each display, only two inverters are required to properly decode the display addresses. Pin 9 is used to reset the 74LS74 flip-flop. When the output of pin 9 goes high, the 74LS74 flip-flop is cleared, inhibiting the clock output from the 555 timer and clearing the 74LS393 counter.

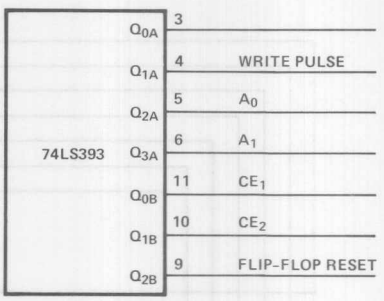


Figure 32. Organization of the 74LS393 for the Circuit Shown in Figure 30

Hex Address	Hex Data	User Message (ASCII Data)	User Message (Comments)
000	50	P	Far Right Character First Message
001	55	U	
002	20		
003	53	S	
004	49	I	
005	20		
006	4D	M	
007	45	E	
008	54	T	
009	53	S	
00A	59	Y	
00B	53	S	
00C	20		
00D	45	E	
00E	48	H	
00F	54	T	Far Left Character First Message
010			Far Right Character Second Message
.			
:			
3F0			Far Right Character Last Message
.			
:			
3FF			Far Left Character Last Message

Figure 33. ROM Organization for the Circuit Shown in Figure 30 Using the Address Decoding Circuit Shown in Figure 14a

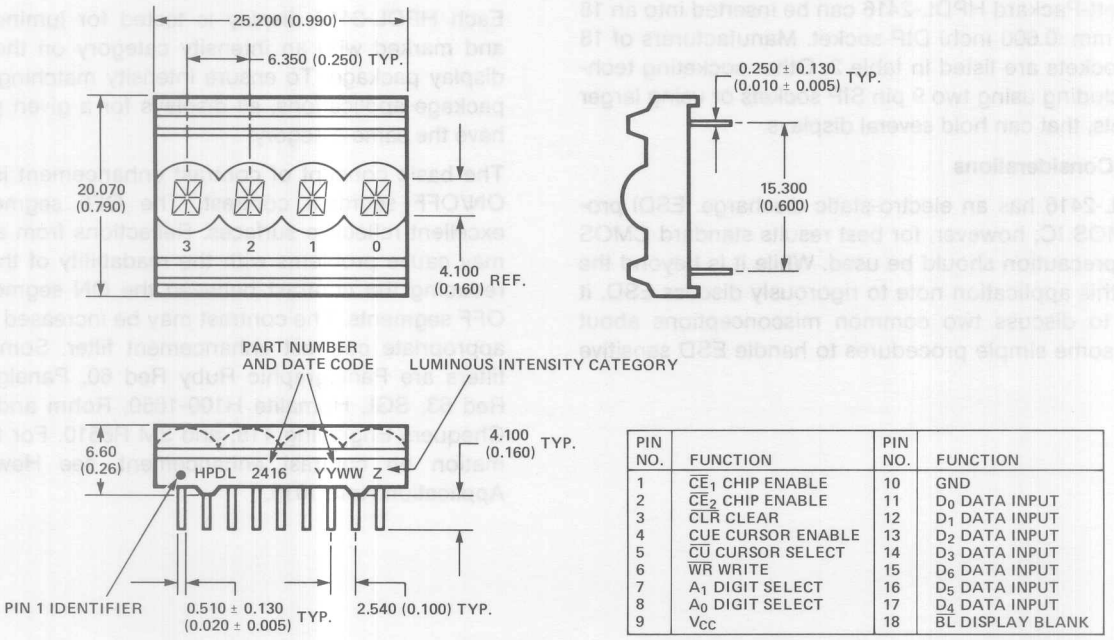
Hex Address	Hex Data	User Message (ASCII Data)	User Message (Comments)
000	54	T	Far Left Character First Message
001	48	H	
002	45	E	
003	20		
004	53	S	
005	59	Y	
006	53	S	
007	54	T	
008	45	E	
009	4D	M	
00A	20		
00B	49	I	
00C	53	S	
00D	20		
00E	55	U	
00F	50	P	Far Right Character First Message
010			Far Left Character Second Character
.			
.			
03F			Far Left Character Last Message
.			
.			
3FF			Far Right Character Last Message

Figure 34. ROM Organization for the Circuit Shown in Figure 30 Using the Address Decoding Circuit Shown in Figure 14b

The organization of the 2716 EPROM depends on the display format. If the display is formatted as shown in Figure 14a, the EPROM is organized as shown in Figure 33. However, if pins 5 and 6 of the 74LS393 counter are inverted, the display is formatted as shown in Figure 14b and the EPROM is organized as shown in Figure 34. The 74LS273 latch selects the message to be displayed through the higher order address inputs of the EPROM.

MECHANICAL AND ELECTRICAL HANDLING

The HPDL-2416 can be stacked horizontally and vertically to create messages of any size. Figure 35 shows the package dimensions for the part. The display is designed to operate continuously from –20° C to +70° C for all possible input conditions (including the illuminated cursor in all four character locations).



PIN NO.	FUNCTION	PIN NO.	FUNCTION
1	$\overline{CE}_1$ CHIP ENABLE	10	GND
2	$\overline{CE}_2$ CHIP ENABLE	11	D_0 DATA INPUT
3	$\overline{CLR}$ CLEAR	12	D_1 DATA INPUT
4	CUE CURSOR ENABLE	13	D_2 DATA INPUT
5	$\overline{CU}$ CURSOR SELECT	14	D_3 DATA INPUT
6	$\overline{WR}$ WRITE	15	D_6 DATA INPUT
7	A_1 DIGIT SELECT	16	D_5 DATA INPUT
8	A_0 DIGIT SELECT	17	D_4 DATA INPUT
9	V_{CC}	18	$\overline{BL}$ DISPLAY BLANK

NOTES:
1. UNLESS OTHERWISE SPECIFIED, THE TOLERANCE ON ALL DIMENSIONS IS 0.254 mm (0.010 IN.)
2. DIMENSIONS IN mm (INCHES).

Figure 35. Package Dimensions

Soldering and Post Solder Cleaning

The HPDL-2416 may be hand soldered or wave soldered with SN63 solder. Hand soldering may be safely performed only with an electronically temperature controlled soldering iron that is securely grounded. For best results, the iron tip temperature should be set at 315°C (600°F). For wave soldering, a rosin based RMA flux or a water soluable organic acid (OA) flux can be used. The solder wave temperature should be 245°C ±5°C (473°F ±9°F), and the dwell in the wave shoud be between 1 1/2 to 3 seconds for optimum soldering. Preheat temperature should not exceed 93°C (200°F) as measured on the solder side of the PC board.

Post solder cleaning may be performed with a solvent or aqueous process. For solvent cleaning, Allied Chemical Genesolv DES, Baron Blakeslee Blaco-Tron TES or Du Pont Freon TE can be used. These solvents are azeotropes of trichlorotrifluroethane FC-113 with low concentrations of ethanol (5%). The maximum exposure time in the solvent vapors at boiling temperature should not exceed 2 minutes. Solvents containing high concentrations of alcohols, pure alcohols, isopropanol or acetone should not be used as they will chemically attack the nylon lens. Solvents formulated with trichloroethane FC-111 or FC-112, and trichloroethylene (TCE) are not recommended.

An aqueous cleaning process is highly recommended. A saponifier, such as Kester Bio-kleen Formula 5799 or equivalent, may be added to the wash cycle of an aqueous process to remove to remove rosin flux residues. Organic acid flux residues must be thoroughly removed by an aqueous cleaning process to prevent corrosion of the leads and solder connections. The optimum water temperature is 60°C (140°F). The maximum cumulative exposure if the HPDL-2416 to wash and rinse cycles should not exceed 15 minutes.

Socketing Considerations

The Hewlett-Packard HPDL-2416 can be inserted into an 18 pin 15.24 mm (0.600 inch) DIP socket. Manufacturers of 18 pin DIP sockets are listed in table 2. Other socketing techniques including using two 9 pin SIP sockets or using larger DIP sockets, that can hold several displays.

Handling Considerations

The HPDL-2416 has an electro-static discharge (ESD) protected CMOS IC; however, for best results standard CMOS handling precaution should be used. While it is beyond the scope of this application note to rigorously discuss ESD, it is useful to discuss two common misconceptions about ESD and some simple procedures to handle ESD sensitive devices.

The first misconception is that only unmounted components are susceptible to ESD. Unless a circuit board is properly protected, mounting the component in a PC board can increase the risk of ESD damage. The traces allow the discharge to be conducted to more than one part and provide additional discharge points.

The second misconception is that only low humidity environments allow the production of ESD. While the fact is true an increase in the humidity will lower the ESD voltage, it does NOT eliminate ESD. The average person will not perceive and ESD voltage of less than 3 kV to 4 kV. The voltages encountered during high humidity are typically less than 3 kV to 4 kV, and damage to the part can occur at these lower voltages.

Control of ESD prevention centers on the following basic rules:

Rule No. 1: Treat all electronic parts and assemblies as static sensitive.
Rule No. 2: Handle all sensitive parts and assemblies at "Static-Safe work stations."
Rule No. 3: Package parts properly for storage or transportation by using conductive foam, conductive tubes or conductive bags.

Table 2. Socket Manufacturers

Aries Electronics, Inc. P.O. Box 130 Frenchtown, NJ 08825 (201) 996-6841	Samtec Electronic Hardware P.O. Box 1147 New Albany, IN 47150 (812) 944-6733

CONTRAST ENHANCEMENT

Each HPDL-2416 display is tested for luminous intensity and marked with an intensity category on the side of the display package. To ensure intensity matching for multiple package applications, all displays for a given panel should have the same category.

The basic concept of contrast enhancement is to improve ON/OFF segment contrast. The OFF segments provide excellent reflective surfaces. Reflections from ambient light may cause problems with the readability of the display by reducing the contrast between the ON segments and the OFF segments. The contrast may be increased by using the appropriate contrast enhancement filter. Some suggested filters are Panelgraphic Ruby Red 60, Panelgraphic Dark Red 63, SGL Homalite H100-1650, Rohm and Haas 2423, Chequers Engraving 118, and 3M R6510. For further information on contrast enhancement, see Hewlett-Packard Application Note 1015.